(Часть 2.1) Практические задачи

Для данных задач в гитах проекта мавен используйте src\main\java\stage2\practice\one

1. **Парсер**

   Задана строка, состоящая из символов «(», «)», «[», «]», «{», «}». Проверить правильность расстановки скобок.

   Использовать структуру стек.

   Пример правильной строки:

   *(abc * (a)) - [a/c+{a,b,c}]*

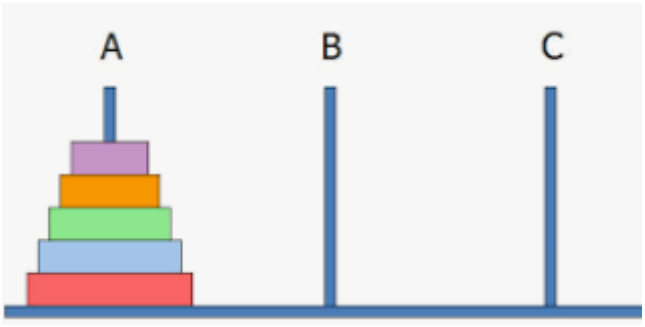
   Пример НЕ правильной строки:

   (abc(a)-{[a/c+a,b,c}]* нет закрывающей), не верно открыта/закрыта фигурная скобка.

2. **Ханойская башня**

   Дано:

   три высоких вертикальных столбика-башни A, B и C. Диски с отверстиями нанизаны на башню A. Самый широкий диск — D1 — находится внизу. Остальные диски, расположенные над ним, обозначены возрастающими цифрами и постепенно уменьшаются кверху D2, D3, D4... (детская пирамидка).

   Цель:

   переместить все диски с башни A на башню C, учитывая ограничения:

   - Диски можно перемещать только по одному.
   - Единственный доступный для перемещения диск — тот, что расположен наверху любой башни.
   - Более широкий диск никогда не может располагаться поверх более узкого.

   Использовать структуру стек.

На вход в программу необходимо задать количество дисков. Работу алгоритма выводить пошагово. То есть действия алгоритма необходимо отобразить. Например:
[1, 2, 3][][]
[1, 2][][3]
[1][2][3]
...
[][][1, 2, 3]

3. Архиватор

Предисловие. Как известно, в java размерность char равняется 16 битам (по умолчанию Java использует кодировку UTF-16) и поэтому может содержать значения от 0 до 65535. То есть в диапазон попадает много символов, знаков, алфавитов.

Допустим, имеем свой алфавит из 4 букв A, C, G, T. Как указано выше каждый символ в java это 16 бит. Зачем так много? У нас всего 4 символа! Мы знаем, что в двоичном коде для хранения типа с четырьмя возможными значениями требуется всего 2 бита! Двоичные значения 00, 01, 10 и 11 — это четыре различных значения, которые могут быть представлены 2 битами. Соответственно представим:

- A - 00 C — 01 G — 10 T — 11

Легко подсчитать, что объем хранилища, необходимого для хранения строк с таким представлением, может быть уменьшен более чем на 87% (с 16 до 2 битов).

Задание. Сделать архиватор с функциями compress (упаковка) строк из символов данного алфавита и decompress (распаковка) из битового представления в строку. Для примера: TAGG - 11001010
Для перевода в битовое представление и назад **ОБЯЗАТЕЛЬНО** использовать BitSet. Полученное в BitSet битовое представление вывести на экран для демонстрации результата.